

June 10th, 2010

# H\_DigiPLC

## User guide

History :	2
General information:	2
Main window:	3
Settings:	4
Communication :	5
Digibutler access (TCP/IP):	5
Serial:	6
File Mappings :	6
Datalogger :	7
PLC :	8
Modification of parameters :	9
Transfert :	10
Disconnected :	10
Connected :	11
DataLogger :	12
Disconnected :	12
Connected :	13
Reading Configuration Datalogger :	14
Common Config :	14
Events Config :	15
Writing configuration Datalogger :	15
Display records :	16
Ladder :	18
Creation :	18
Tools :	19
Extensions :	19
Rules:	20
Operating mode :	21
Compiling :	26
Running :	29
Monitoring :	31
Performances :	32
Compatibilities Platforms Digibutler, ARMEE and Sceptre:	33
Differences	33
Communication PC ⇔ Target	33
Transfert files on SD card	33
Datalogger	34
PLC	34

## History :

Date	Version	Author	Description	
June 1st, 2009	1.0	Henri Laidet	Creation	
June 25th, 2009	1.1	Henri Laidet	Addition actions on integers (move, add ...)	
Sept 4th, 2009	1.2	Henri Laidet	Addition communication with ARM7 (serial com port)	
Feb 1 st, 2010	1.3	Henri Laidet	Addition compatibility DigibutlerK3 and MQX	
June 12 st, 2010	1.4	Henri Laidet	Adaptation serial COM via Bluetooth (sceptre)	

## General information:

The program is written with « *Microsoft Visual C# 2008 Express Edition* ».

H\_DigiPLC is of use to the dialogue between PC and DigiButler ( DB).

Since version 1.2, the dialog is also made between PC and ARMée via a port com.

Since version 1.4, dialog runs with Bluetooth serial profil (sceptre ARM7)

To exchange files, use Datalogger, create and execute PLC programs.

### Digibutler :

ColdFire MCF52231

1x Ethernet RJ45

1x Serial com port RS232 DE9

### ARMée :

ARM7TDMI LPC2106

1x Serial com port RS232 DE9

1x Serial com port RS232 or RS485 DE9

### Sceptre :

ARM7TDMI LPC2148

1x Serial com FTDI USB

1x Serial com Bluetooth

## Main window:

MDI Environment

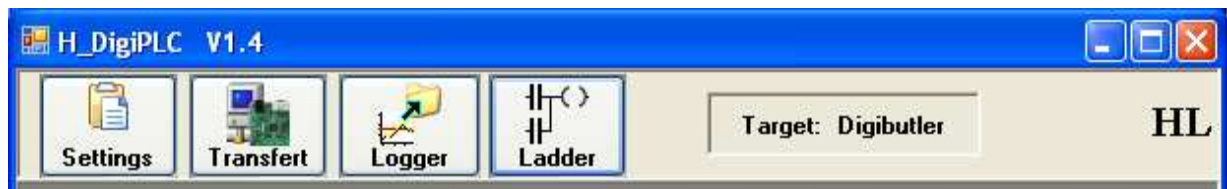


Figure 1

## Settings:

Configuration: assignment and saving

C# does not allow the access the registry, everything is in the file « *HDB\_settings.xml* »

Fields T1 and T2 (Figure 2) allow to change target

- Field T1: Target Digibutler is use to coldfire specificities (8 bits ports, Big-Endian)
- Field T2: Target ARM7 idem T1 but for ARM7 (32 bits port, Little-Endian)

The screenshot shows a 'Settings' dialog box with a blue title bar and a close button. The 'Target' section has two radio buttons: 'Digibutler (CFire MCF5223x-5x)' (selected, labeled T1) and 'ARM7 (LPC2148)' (labeled T2). Below is a tabbed interface with 'Communication', 'File Mappings', 'Datalogger', and 'PLC' tabs. The 'Communication' tab is active, showing two radio buttons: 'TCP/IP' (selected, labeled C1) and 'Serial' (labeled C2). Under 'Digibutler access (TCP/IP)', there are three text input fields: 'Address (IP or name)' with '192.168.1.152' (labeled 1), 'Port (HTTP)' with '80' (labeled 2), and 'Key' with 'joshua' (labeled 3). A checked checkbox labeled 'Add "\n" at the end of command ( required for DigibutlerK3 and MQX )' is labeled 4. The 'Serial' section is disabled and contains dropdowns for 'COM Port' (COM40), 'Baud Rate' (115000), 'Parity' (None), 'Data Bits' (8), and 'Stop Bits' (One). A 'Close' button is at the bottom.

Figure 2

## Communication :

Les champs C1 et C2 (Figure 2) fixent le type communication

- Champ C1 : Communication en TCP/IP (Digibutler)
- Champ C2 : Communication via port série (ARM7)

Fields C1 and C2 (Figure 2) define the communication type

- Field C1: Communication by TCP/IP ( Digibutler)
- Field C2: Communication via serial com port ( ARM7)

### Digibutler access (TCP/IP):

Configuration of the dialogue with DB

- Field 1 : Digibutler IP ( *192.168.1.151* ou *monadresse.dyndns.org* ).
- Field 2 : Port HTTP server
- Field 3 : Key for access to the Tunnel FTP. Allows the navigation in DB files  
[ `ipport.h: #define UPLOAD_KEY "joshua"` ]
- Field 4 : Dialog with DigibutlerK3 ( HTTP server from MQX)

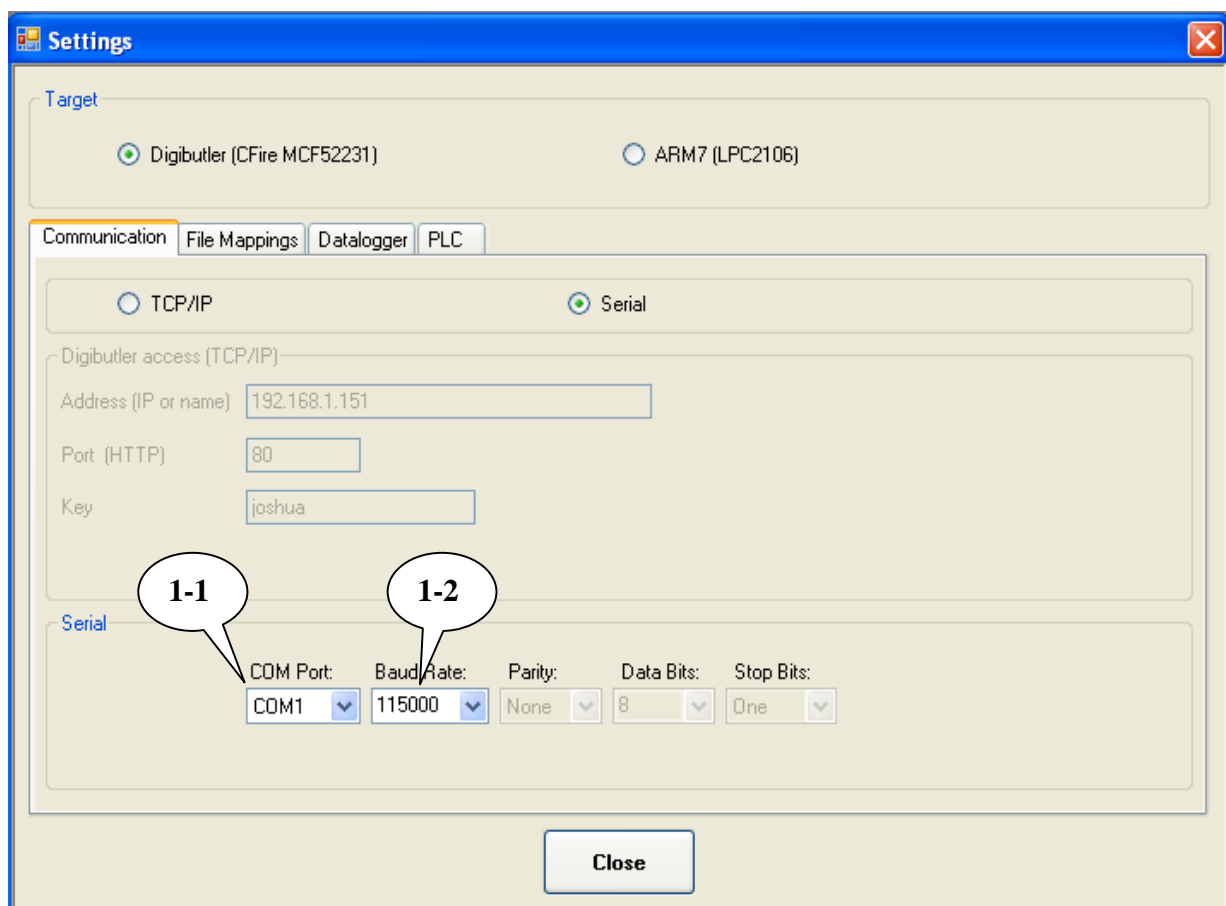


Figure 2-1

## Serial:

Configuration of the communication on serial port (Figure 2-1)

- Field 1-1: Selection of the com port
- Field 1-2: Select baudrate

Parity, data bits and stop bits are imposed and are not modifiable.

## File Mappings :

Association of the extensions of files with an exe file.

Allows to open files DB or PC. (In the function **Transfert** button **Show**).

- Field 5 : List of the extensions associated with a type of file ( Field 6 ).
- Field 6 : Type of files (5 in everything IMAGE, TEXT, EXCEL, PDF, free ).
- Field 7 : Associated extension
- Field 8 : exe file connected to the extension Field 7
- Field 9 : Add a group (Type, Extension, exe file) in the list Field 5
- Field 10 : Delete an element of the list Field 5

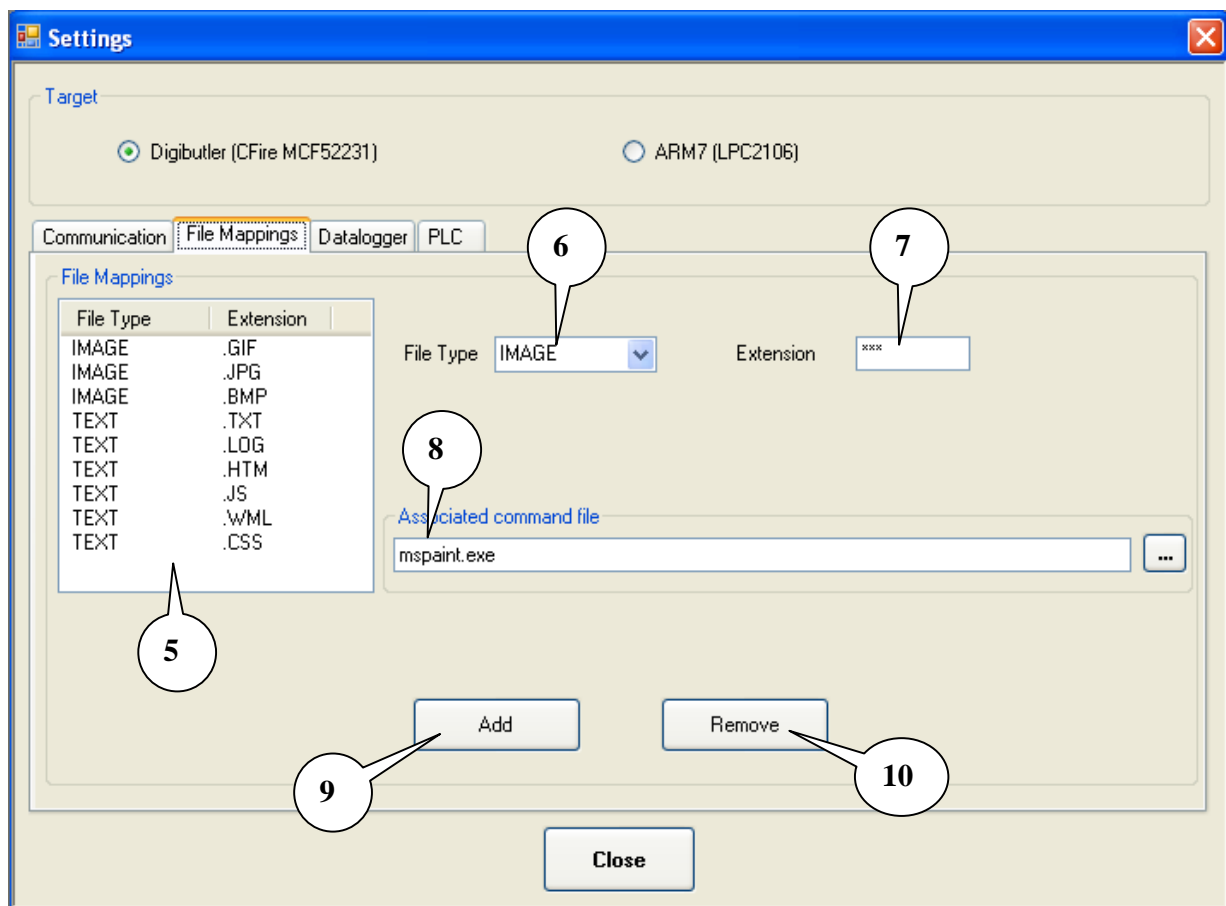


Figure 2-2

## **Datalogger :**

Configuration of the access to Datalogger (Figure 2-3)

- Field 11 : Name of the file of configuration on DB  
[ `Datalogger.c: #define DATA_LOGGER_CONF_NAME "config.cfg"`]
- Field 12: Offset of correction the hour with regard to the GMT

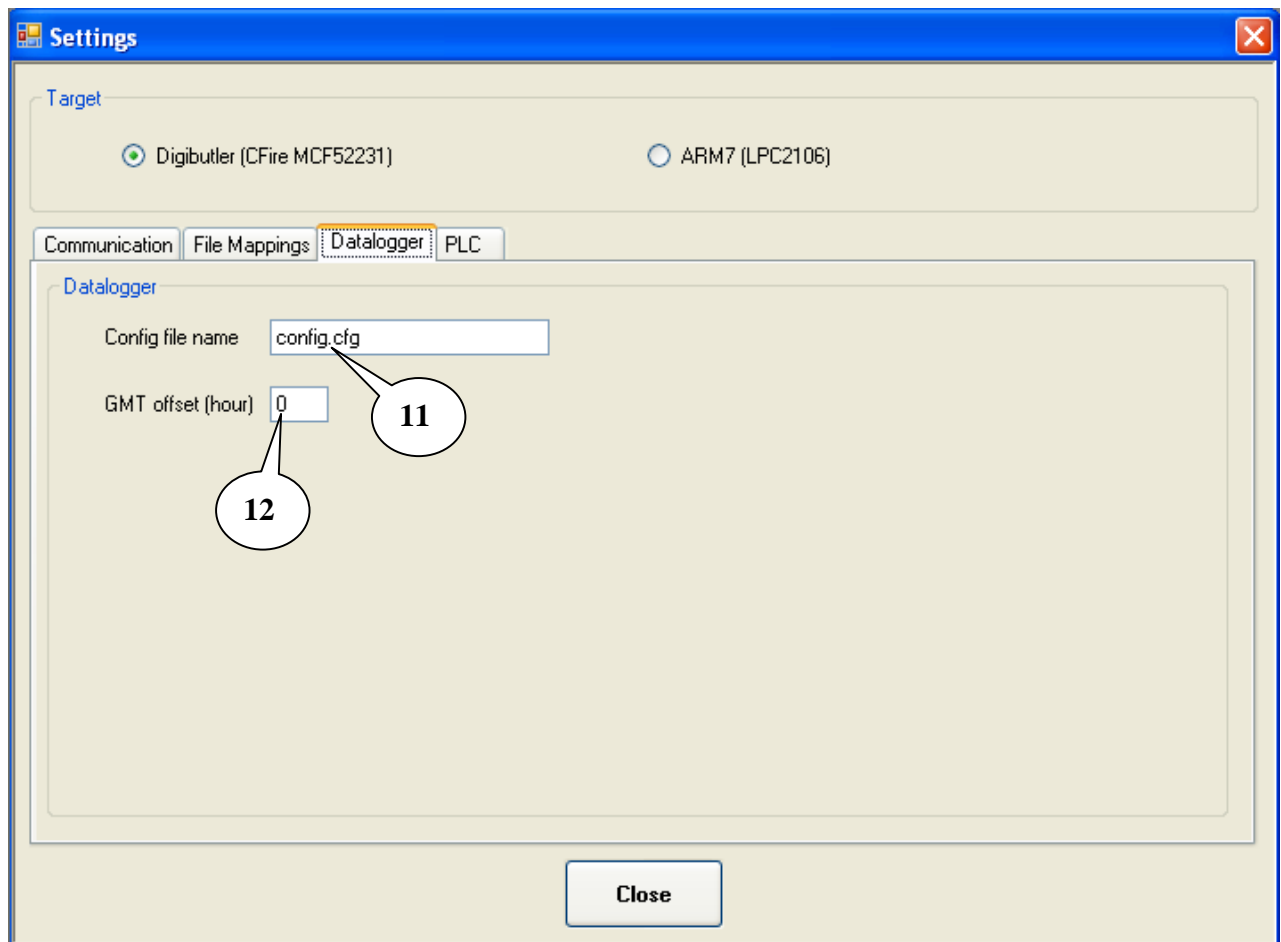


Figure 2-3

## PLC :

Configuration of the PLC (Figure 2-4)

- Field 13 : Name of the file on DB. It is the useful compiled file by the engine.  
[ PLC.h: #define PLC\_FILENAME "plc.txt"]
- Field 14: Name of the graphic file on the PC. It is the file source of Ladder.
- Field 15: Main port: mask bits used as IO TOR  
8 bits (GPT0-3) on Digibutler ou 32 bits (port0) on ARM7
- Field 16 : Second port : mask bits used as IO TOR  
8 bits (ANA inputs) on Digibutler or non-existent on ARM7 ( LPC2106)
- Field 17 : Number of expanders (used to propose the list of the IO TOR).
- Field 18 : Number of Bits of general usage.
- Field 19 : Number of Temporizations.
- Field 20 : Number of integers ( 32 signed bits) of general usage.

*During the connection PC ⇔ PLC, if Fields 15 - 20 does not correspond, there is demand of automatic update..*

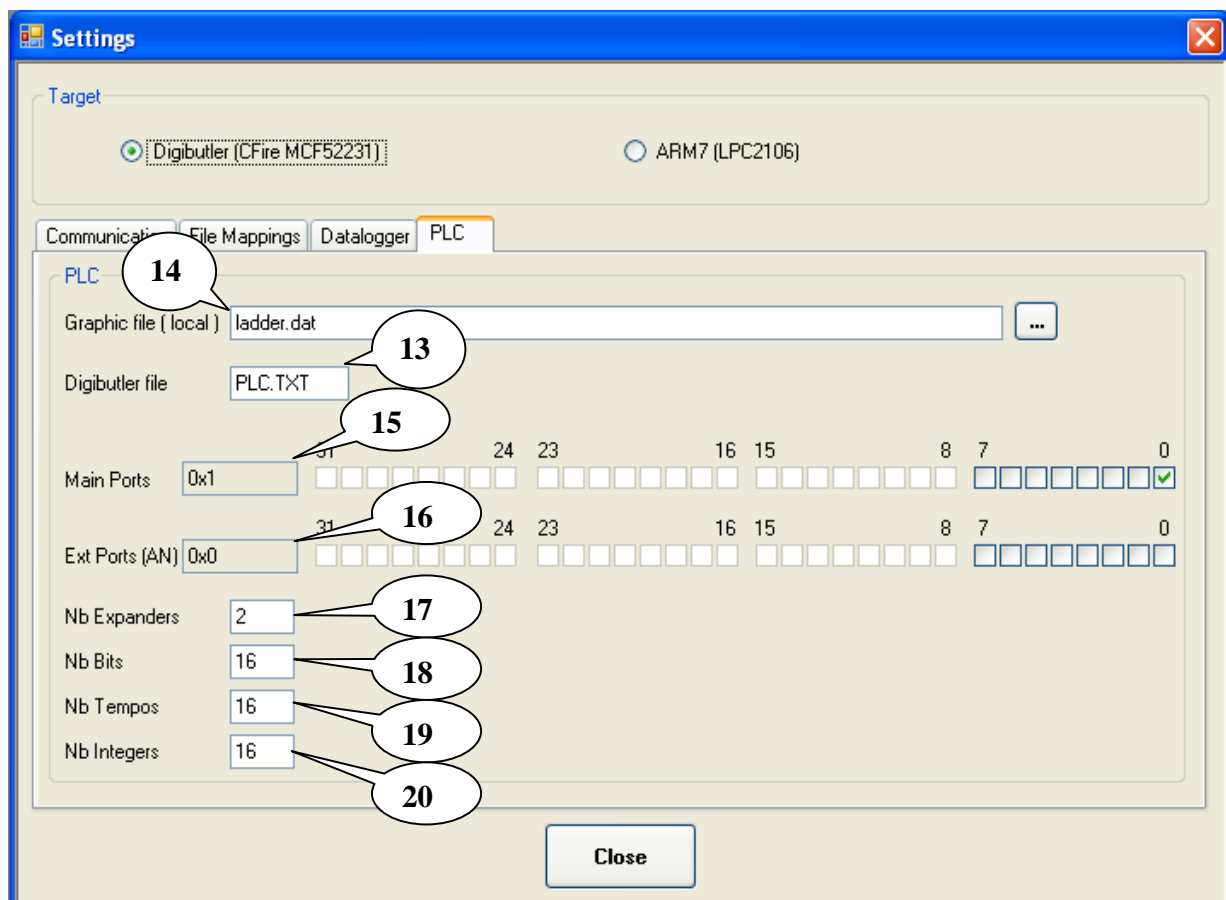


Figure 2-4



***Modification of parameters :***

**If we modify a parameter, when we leave the program we have a message**



Figure 3

The answer Yes backs up the file XML before leaving.

## Transfert :

This feature is implemented on DigiButler and ARM7.

Allows to read and to write the files on target (stored in SD card).

## Disconnected :

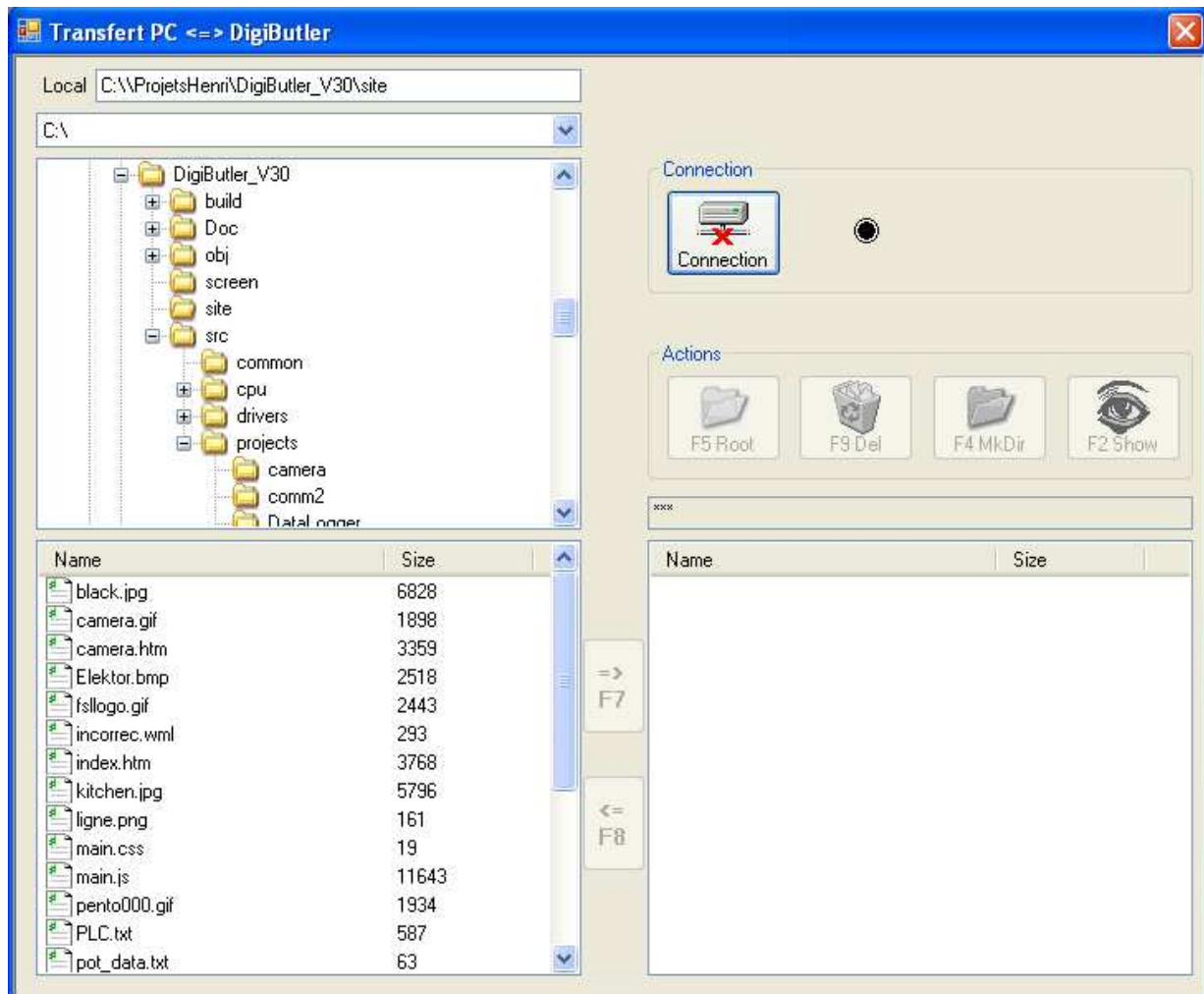


Figure 4

Left half is the arborescence of the PC.

A modification of the disk or the current directory is saved with *Settings* to show the same path in the next run of the program.

The message of modification will thus appear.

All the buttons are disabled except that of connection with DB.

## Connected :

If **Settings** Fields 1, 2 and 3 are correct, the connection has to be correctly made and show :

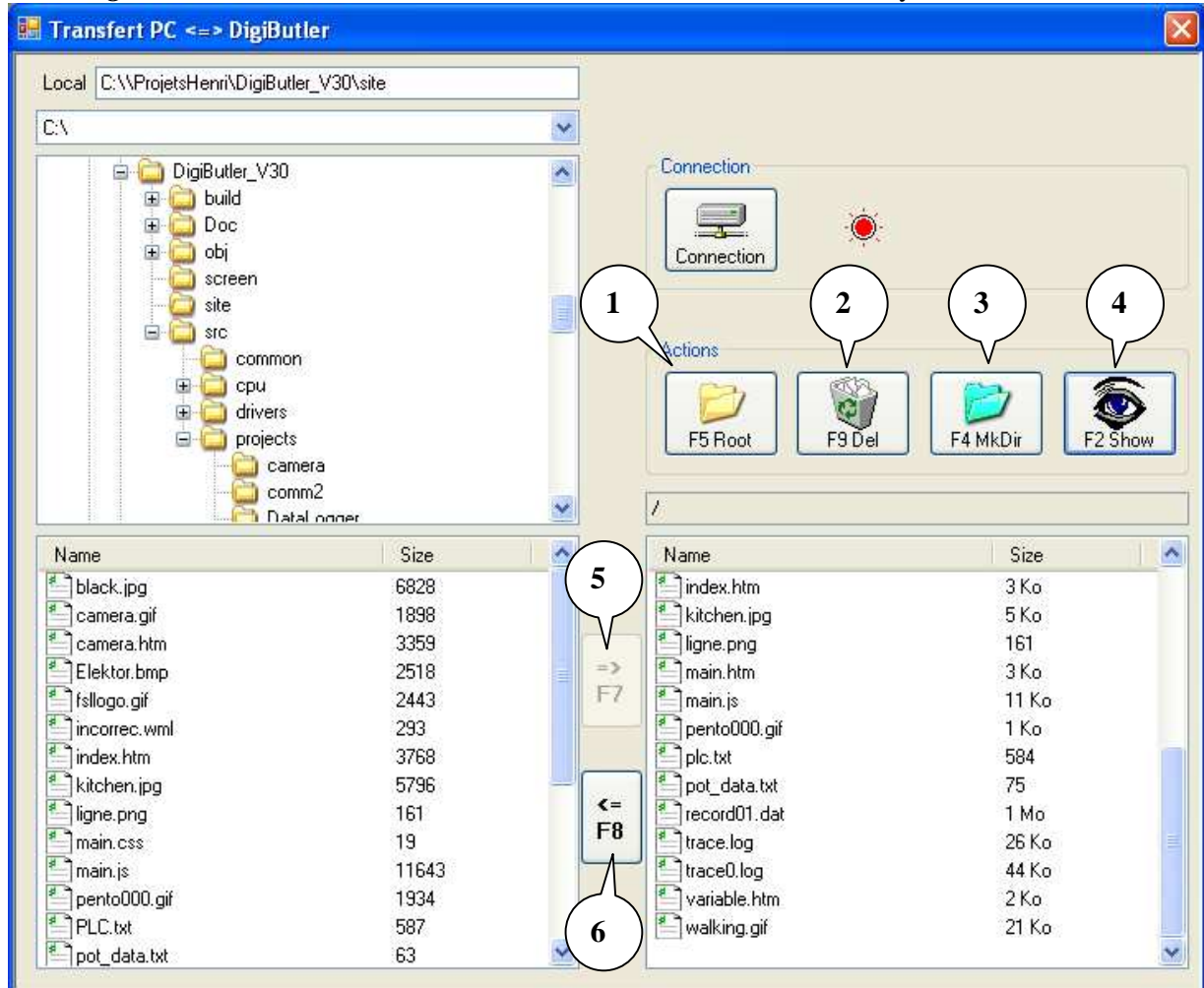


Figure 5

- Button 1 : Return to the root (when there are directories)
- Button 2 : Delete selected file (on DB)
- Button 3 : Creation of a directory (on DB)
- Button 4 : Load the selected file with the exe file (**Settings File Mappings**)
- Button 5 : Transfer the selected file from PC to DB
- Button 6 : Transfer the selected file from DB to PC

Button 4 appears when the selected file has a known extension.

Button 5 appears when the selected file is of highly-rated PC.

Button 6 appears when the selected file is of highly-rated DB.

## DataLogger :

Feature not implemented on ARM7.

Useful only with Digibutler ( ColdFire).

### Disconnected :

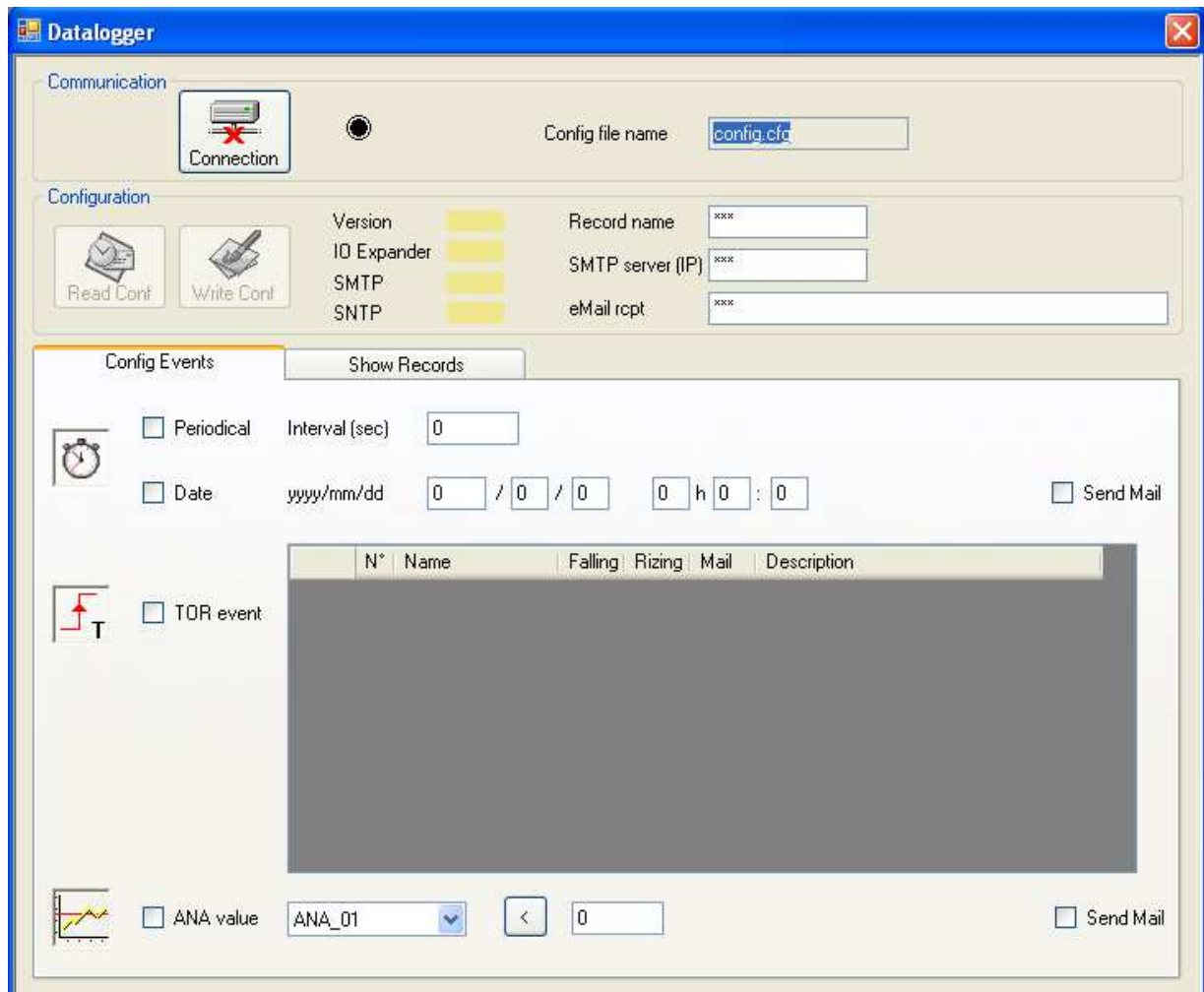


Figure 6

### **Connected :**

After PC connected to DB, Buttons **Read Conf** and **Write Conf** become valid.

The version of the soft Datalogger, the number of expanders are shown.

And if SMTP client and SNTP client are activated on DB.

The SMTP client allows to send emails on performed condition.

The SNTP client updates RTC and thus allows to use an event in date.4

The screenshot shows the 'Datalogger' software window. It has three main sections: Communication, Configuration, and Config Events.

**Communication:** Includes a 'Connection' button and a 'Config file name' field set to 'config.cfg'.

**Configuration:** Includes 'Read Conf' and 'Write Conf' buttons. It displays the following settings:

- Version: 2.0
- ID Expander: 2
- SMTP: OFF
- Sntp: ON
- Record name: xxx
- SMTP server (IP): xxx
- eMail rcpt: xxx

**Config Events:** Includes a 'Show Records' button and several event configuration options:

- Periodical:** Interval (sec) set to 0.
- Date:** Format yyyy/mm/dd with fields for year, month, day, hour, and minute. A 'Send Mail' checkbox is present.
- TOR event:** Includes a 'T' icon and a 'Send Mail' checkbox.
- ANA value:** Includes a graph icon, a dropdown menu set to 'ANA\_01', a '<' button, and a value field set to 0. A 'Send Mail' checkbox is present.

**Event Table:**

N°	Name	Falling	Rizing	Mail	Description
1	GPT0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	GPT1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	GPT2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	GPT3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
5	EXP1-0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
6	EXP1-1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
7	EXP1-2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
8	EXP1-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 7

## Reading Configuration Datalogger :

The Button *Read Conf* reads the file of configuration stored in DB.

### Settings Datalogger Field 11

The screenshot shows the 'Datalogger' configuration window. It is divided into several sections: 'Communication', 'Configuration', 'Config Events', and a table of events. Numbered callouts point to specific fields:

- 1: 'Config file name' text box containing 'config.cfg'.
- 2: 'Record name' text box containing 'record01.dat'.
- 3: 'SMTP server (IP)' text box containing '193.252.23.66'.
- 4: 'Periodical' checkbox (checked) under 'Config Events'.
- 5: 'Date' checkbox (checked) under 'Config Events'.
- 6: 'TOR event' checkbox (checked) under 'Config Events'.
- 7: 'ANA value' checkbox (checked) under 'Config Events'.

The 'Config Events' section also includes an 'Interval (sec)' field set to 60, a date/time picker set to 2009/06/10 15:30:00, and a 'Send Mail' checkbox (checked). The table below lists various events with their configurations.

N°	Name	Falling	Rizing	Mail	Description
1	GPT0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Relais carte
2	GPT1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	GPT2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	GPT3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
5	EXP1-0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Porte bureau
6	EXP1-1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Fenetre bureau
7	EXP1-2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Test 3
8	EXP1-3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

At the bottom, there is an 'ANA value' section with a dropdown menu set to 'ANA\_01', a '<' button, a '2000' text box, and a 'Send Mail' checkbox (unchecked).

Figure 8

### Common Config :

- Field 1 : Name of the file where are recorded the events ( binary file).
- Field 2 : Address of your SMTP server (sending of E-mails).
- Field 3 : Address of the recipient (sending of E-mails).

### **Events Config :**

- Field 4 : Defines that there is periodic recording (Field interval).
- Field 5 : Defines that there is recording in date (and sending or not of an E-mail).
- Field 6 : Defines that there is recording on change of an IO TOR  
Every IO is configurable to generate a recording on rising edge, falling edge or 2. Furthermore, we can send an E-mail.
- Field 7 : Defines that there is recording when one of 8 analog input becomes lower, upper, equals or different of a value. With sending of E-mail

### ***Writing configuration Datalogger :***

After one of Fields to be modified, we have to write on DB with the Button ***Write Conf*** so that the modification is taken into account.

## Display records :

The tab *Show records* creates the elements of consultation.

The screenshot shows the 'Datalogger' application window. It has three main sections: 'Communication', 'Configuration', and 'Records'.

- Communication:** Includes a 'Connection' button with a server icon and a 'Config file name' field containing 'config.cfg'.
- Configuration:** Includes 'Read Conf' and 'Write Conf' buttons. It also displays settings: Version 2.0, IO Expander 2, SMTP OFF, and SNTP ON. On the right, there are fields for 'Record name' (record01.dat), 'SMTP server (IP)' (193.252.23.66), and 'eMail rcpt' (henri.laidet@gmail.com).
- Records:** This section is active and contains:
  - A 'Numbers' field with the value 0 and a 'Numbers' button.
  - Navigation buttons: 'First', 'Previous', 'Record' (with a dropdown showing 0), 'Next', and 'Last'.
  - An 'Event' section with a date/time picker set to 0 / 0 / 0 0 h 0 : 0. Below this are three buttons: 'Periodic' (with a clock icon), 'Date' (with a clock icon), and 'TOR' (with a red arrow icon).
  - A table with columns 'N°', 'Name', 'State', and 'Description'. The table body is currently empty.
  - On the right side of the table, there are eight input fields labeled 'ANA 0' through 'ANA 7'.

Figure 9



A pressure of Button **Numbers** questions DB to know the number of present recordings in the file (**Common Config Field 1**) current and so allows the navigation.

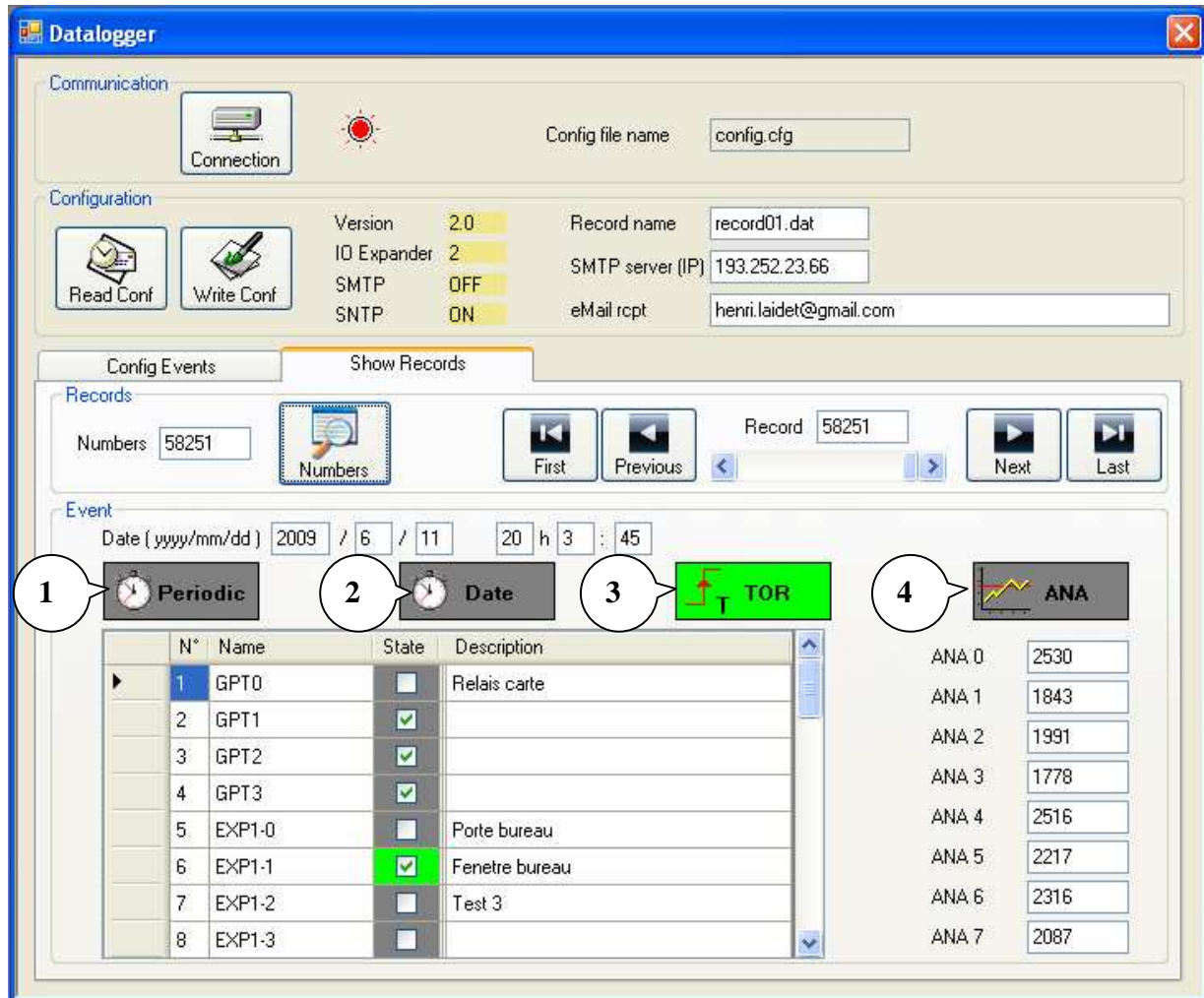


Figure 10

Buttons **First**, **Previous**, **Next** and **Last** move us in the file.

We can also use the slider or directly enter the number of record choose.

The records are dated.

- Field 1 : Green indicates a periodic recording.
- Field 2 : Green indicates a recording in date.
- Field 3 : Green indicates a recording on event TOR.
- Field 4 : Green indicates a recording on analog condition true.

The events can be simultaneous (ex: Periodical and TOR).

**The scan time of Datalogger on DB is 1 second.**

## Ladder :

**This feature is implemented on DigiButler and ARM7.**

On DB (or ARM7), a PLC ( Programmable Logic Controller) engine executes the *plccode* generated here.

The engine is actionable with *#define HL\_PLC 1* (in " ipport.h " on DB or "config.h" on ARM7).

The engine is called every 5ms (a tick). The timebase is thus of 5ms.

The *plccode* is a list of objects (AND, NOW, PUSH, POP) which the engine interprets

H\_DigiPLC allows to edit a program in graphic mode, to compile it, to transfer it for execution on DB (or ARM7) and to debug it (Monitoring).

## Creation :

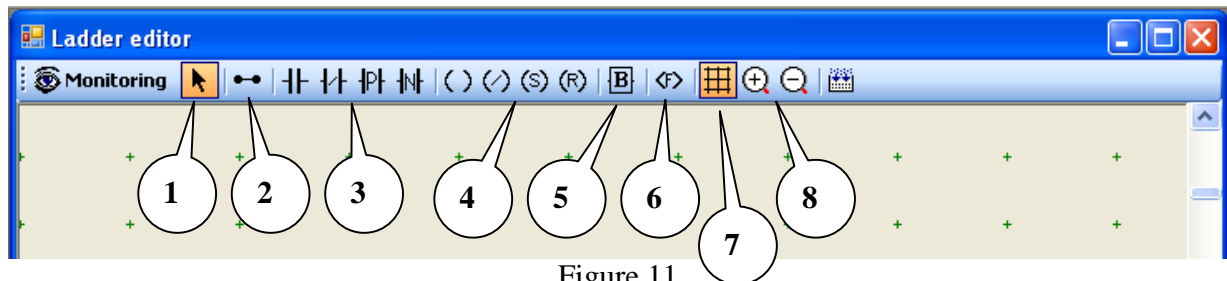














Figure 11

To create a program, there are some tools Fields 1 – 6

- Field 1 : Arrow allows to select an object on the sheet and to move it
- Field 2 : Line allows to connect objects between them
- Field 3 : Inputs (Simple, inverted, rising edge, falling edge).
- Field 4 : Outputs (Simple, inverted, Set, Reset)
- Field 5 : Blocs (special conditions)
- Field 6: Special outputs (no Boolean)
- Field 7 : Show grid on the sheet.
- Field 8 : Enlarges or reduces the graph.

## Tools :

-  Arrow selection and/or movement of objects.
-  Draw segments.  
The grid represents the points of connection between cells.  
With this tool, a small square moves on the grid by following the mouse cursor.  
The 1st click positions the origin of the plan (the square remains spot).  
The second click draw lines between 2 points and begins with the horizontal segment.
-  Simple input: the connection is established if the bit is true.
-  Inverted input: the connection is established if the bit is false.
-  Rising edge: the connection is established if the bit passes from false to true state.
-  Falling edge: the connection is established if the bit passes from true to false state.
-  Simple output: the bit takes the state of entry.
-  Inverted output : the bit takes the inverse state of entry.
-  Set output: if the entry is true the bit is Set else it remains unchanged.
-  Reset output: if the entry is true the bit is Reset else it remains unchanged.
-  Block: function special which do not use boolean. List farther.
-  No bool output: operation on integers (MOVE, ADD, SUB, AND, OR...)

## Extensions :

A right click shows some extensions :

Delete Object	F9
Delete All	F10
Insert Line	Insert
Delete Line	Delete
Refresh	F5
Check equ	F6
Properties	F4

Figure 12

- Delete Object : Delete selected object
- Delete All : Delete all the sheet
- Insert Line : Insert a line in the current position.
- Delete Line : Delete the current line (if empty)
- Refresh : Redraw all the sheet
- Check equ : Check if the part of the graph can be compile.
- Properties : Open the properties of the selected object

## Rules:

The graph is at first converts in boolean equations for every group

Exemple :  $S1 = (\text{bit.bit2}) + \text{bit3}$

### 1. Outputs :

There should be only one output by equation (otherwise impossible to formulate).

The elements of an equation associated with an output must not be connected to another equation (still impossible to formulate).

The output has to be the object most to the right of the graph.

### 2. The lines of supply:

They are not realized by a vertical plan.

That of the left is the left edge of the sheet.

That of the right is connected to the object most to the right of an equation (the output).

Every equation can finish in a different position (they are treated in a separated way).

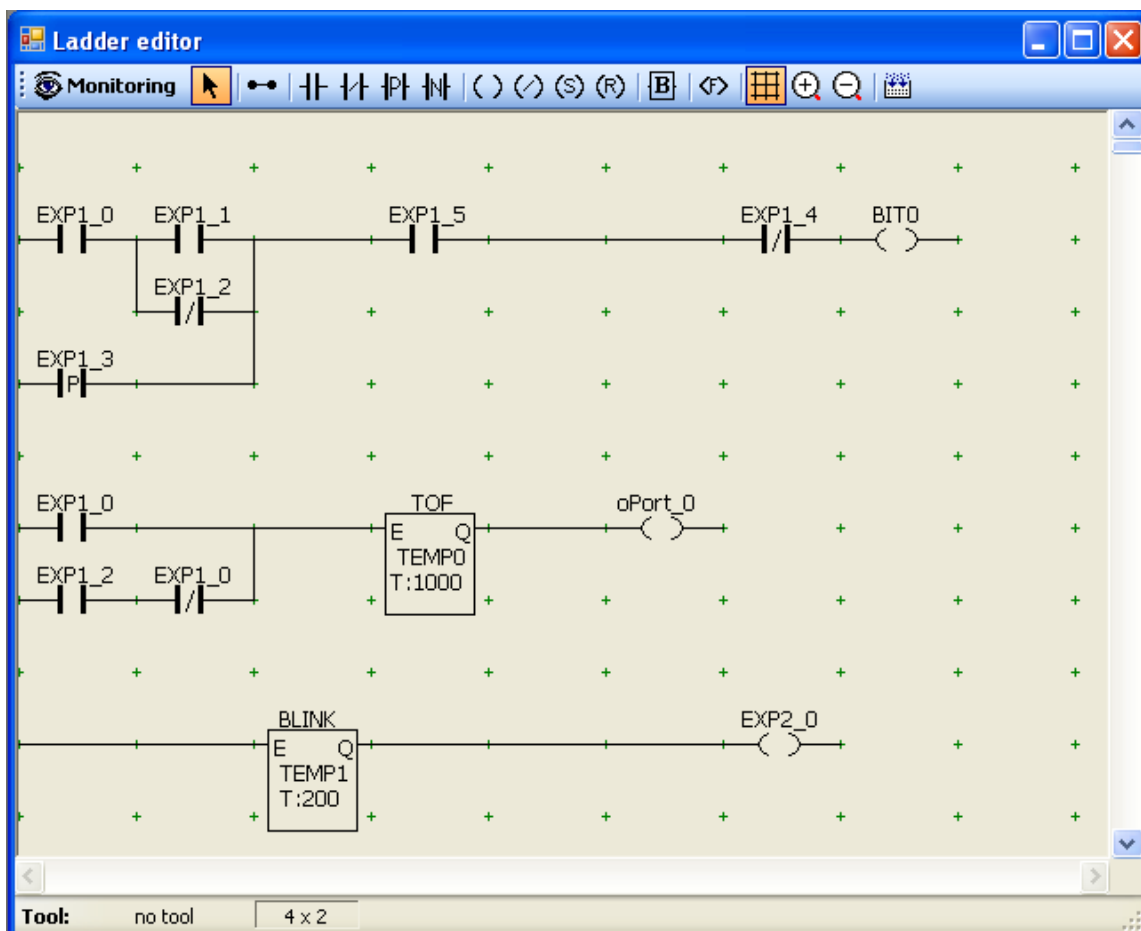


Figure 13

## Operating mode :

We select a tool (type of object ex: Figure 11 Field 3, 4, 5 or 6).

The mouse cursor takes the shape of the selected object

We take place in the targeted cell and click.

A window of properties is shown

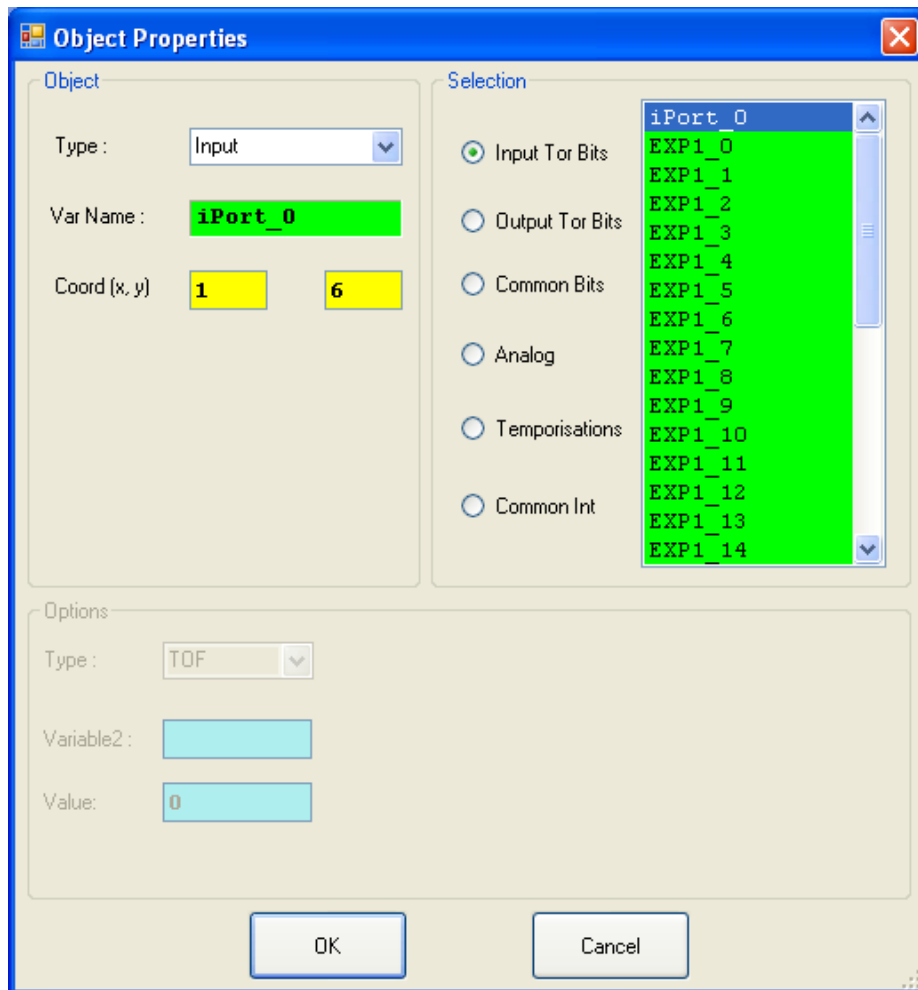


Figure 14

We can choose in the list of selection:

For a contact (simple or not) an input TOR an output TOR or a Bit general usage.

For an output (simple or not) an output TOR or a Bit general usage.

***For Ports, we find iPort\_x and oPort\_x for input and output because on Digibutler, the ports addresses (GPT0-3) are not same according to the direction of the port (MCF\_GPIO\_SETTA for the bits in input and MCF\_GPIO\_PORTTA for the bits to output).***

***On ARM7, we can use indifferently iPort and/or oPort.***

***The PLC engine makes the difference.***

### Blocs are special conditions

There are at present 8 types :

- Bloc TON : Tempo true if E confirm since « time ».
- Bloc TOF : Tempo true if E confirms or E invalidate since « time ».
- Bloc TMP : Tempo true if E confirm no more « time ».
- Bloc BLINK : Flashing

The blocks of temporization have a small graph of chronology.

The timebase is 5ms.

We have to choose in the list of the temporizations.

- Bloc CMP\_LT : Comparison value1 < value2
- Bloc CMP\_EQ : Comparison value1 = value2
- Bloc CMP\_GT : Comparison value1 > value2

Value1 can be an analog input, a temporization or a general Integer usage.

- Bloc HISTER : Becomes true if value1 > (value2+hysteresis)  
Becomes false if value1 < (value2-hysteresis )

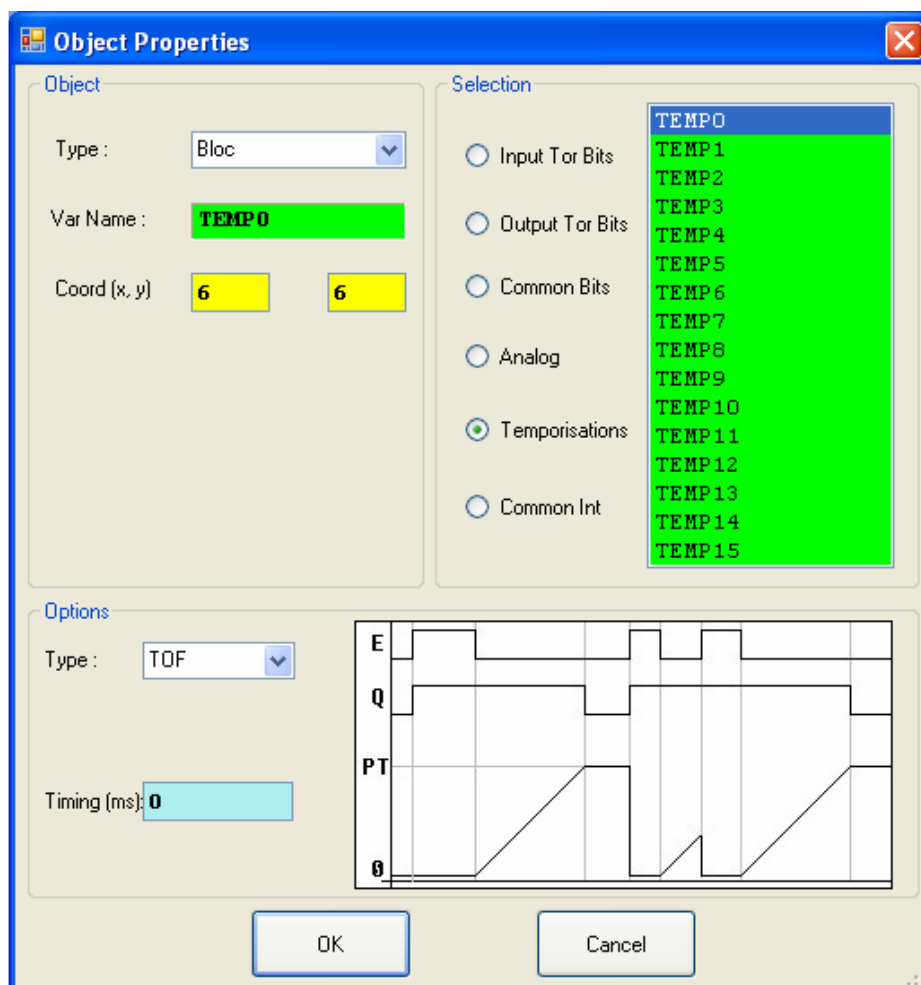


Figure 15

After enter properties (bit, variable, timing), the object is placed on the sheet.

We repeat the operation the number of necessary times.

We use the tool Line to draw the connections between objects.

The tool arrows allows :

1/ to move objects on the sheet

2/ to modify the properties of an object (to change bit, type, timing...)

For that purpose, we select the object (click on the cell) then right click (Figure 12) or F4

### Functions are actions on integers

There are at present 8 types :

- Function MOVE :  $\text{Variable} = \text{variable2}$
- Function ADD :  $\text{Variable} = \text{variable} + \text{variable2}$
- Function SUB :  $\text{Variable} = \text{variable} - \text{variable2}$
- Function AND :  $\text{Variable} = \text{variable and variable2}$
- Function OR :  $\text{Variable} = \text{variable or variable2}$
- Function XOR :  $\text{Variable} = \text{variable xor variable2}$
- Function LSR :  $\text{Variable} = \text{variable} \gg \text{variable2}$
- Function LSL :  $\text{Variable} = \text{variable} \ll \text{variable2}$

The functions have a small graph of help:

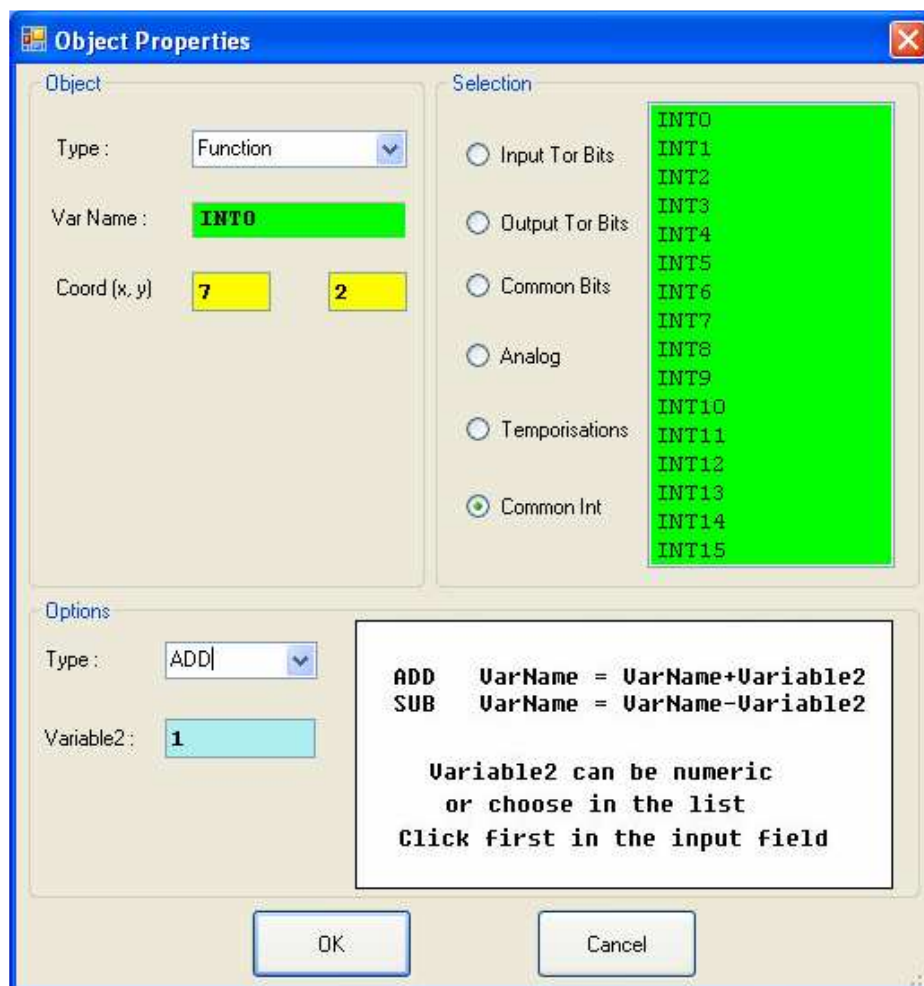


Figure 15-2



To use the functions, DB->PLC has to be at least in V1.1

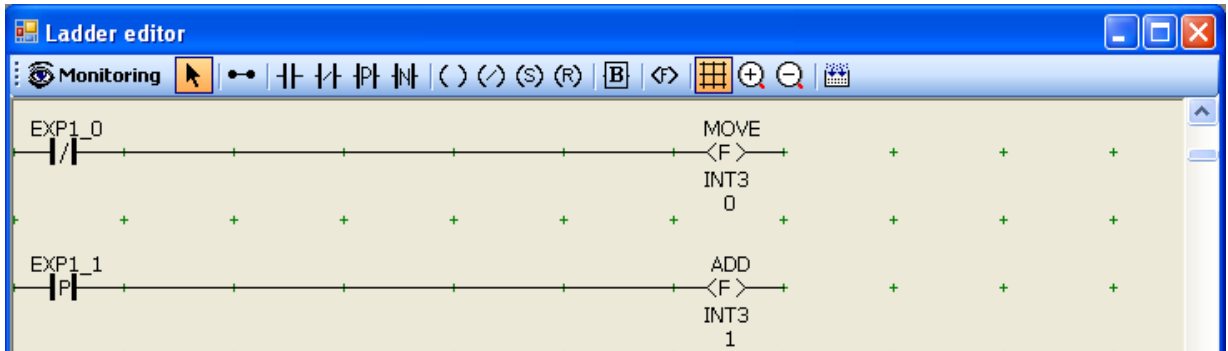


Figure 15-3

## Compiling :

After the creation (or modification) of the graph, a click on **Build PLC file** (Figure 16: Field 1) executes the compilation which opens a window of status.

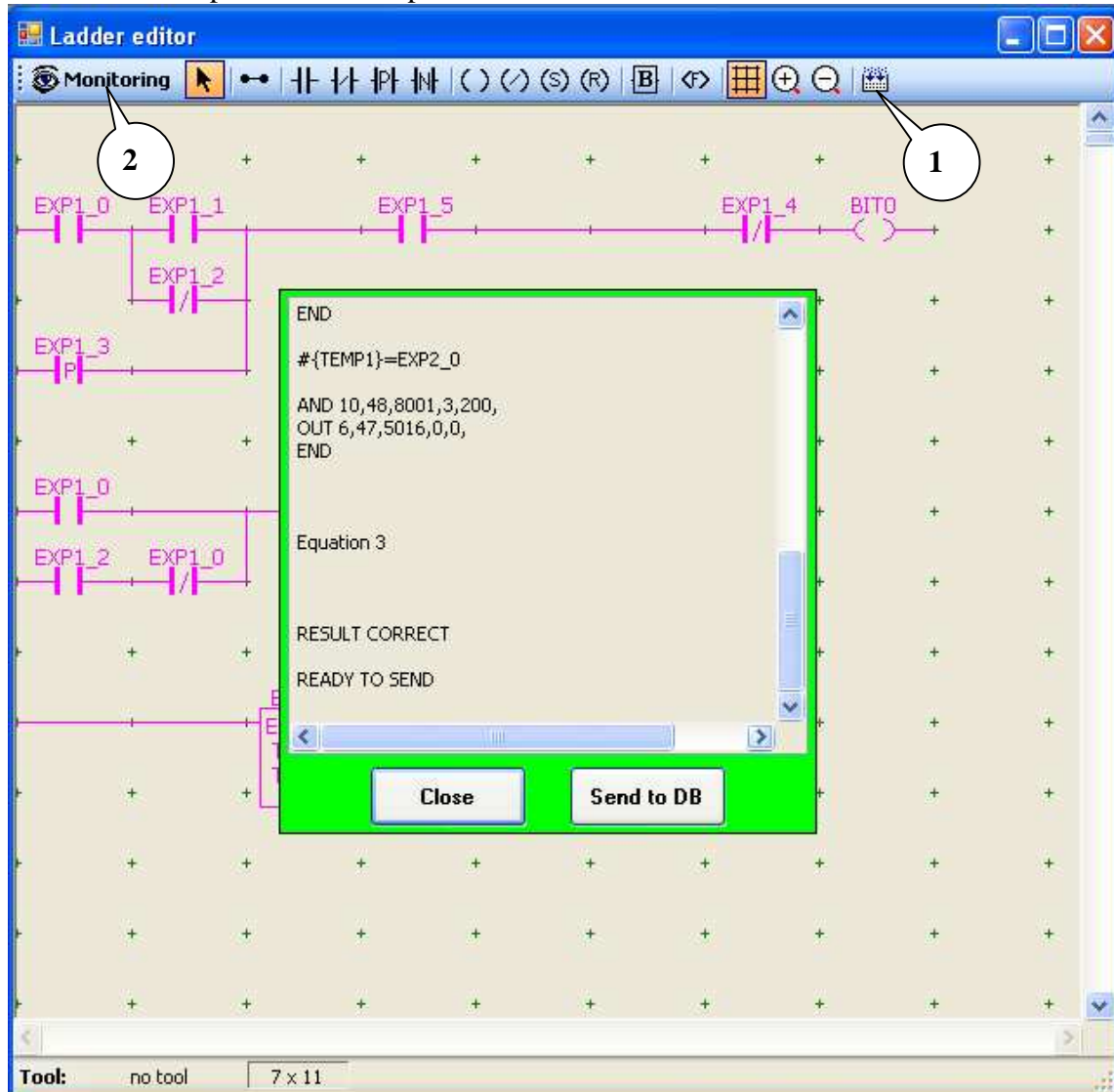


Figure 16

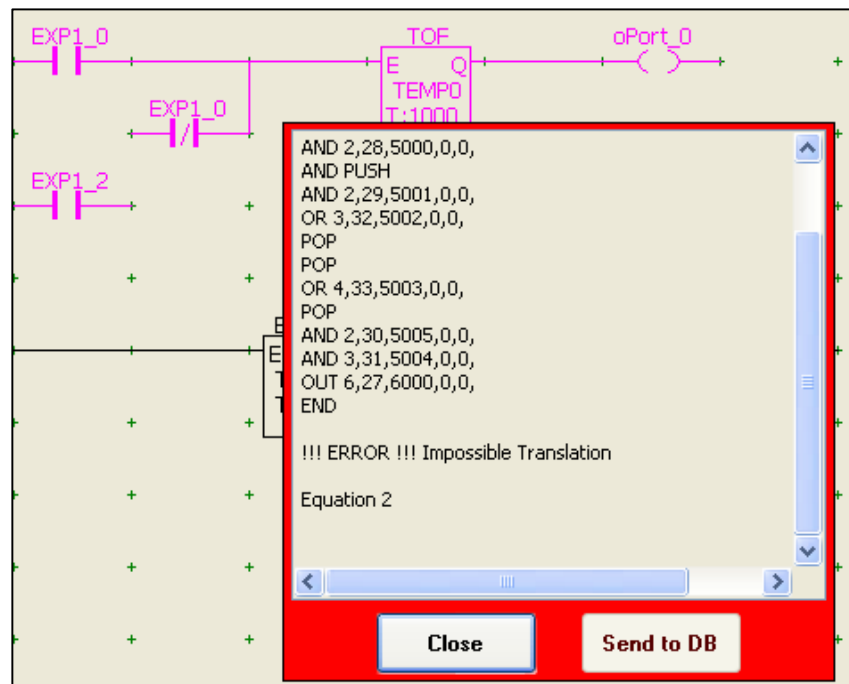


Figure 17

If there is an error, the window of status is red and Button *Send to DB* is disabled (Figure 17). Without error, the window of status is green and Button *Send to DB* is valid (Figure 16).

In case there is no error, a click on *Send to DB* sends the generated file to target (DB or ARM7). The function PLC of DB executes this:

- 1/ Stop the current Ladder program + free of allocated memory.
- 2/ Reading and parsing the file received ( PLC.TXT) to know the necessary RAM.
- 3/ Load the file (parsed) and store in memory.
- 4/ Execution of the program once to measure its timing (in 10 $\mu$ s).
- 5/ Returns size and timing then wait in standby state.

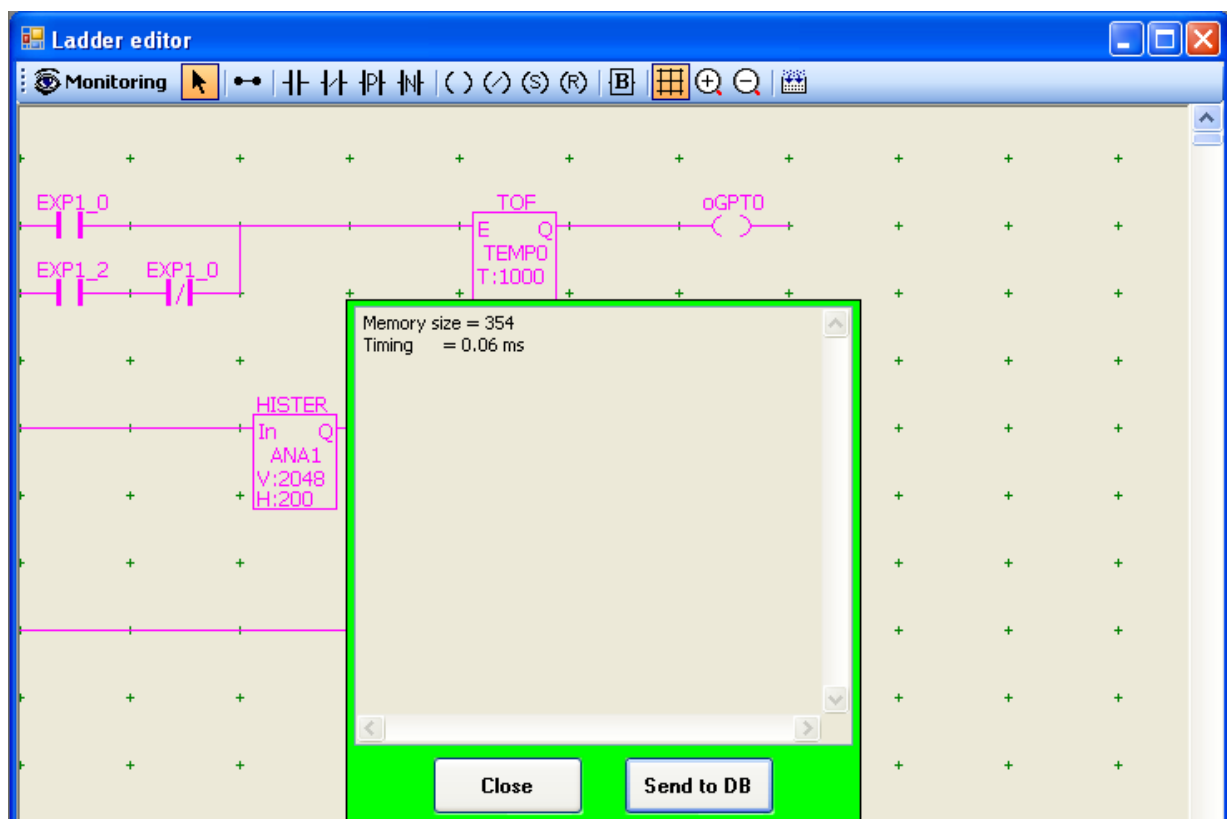


Figure 18

## Running :

A click on **Monitoring** (Figure 16 Field 2) open the control section to the left of the sheet.

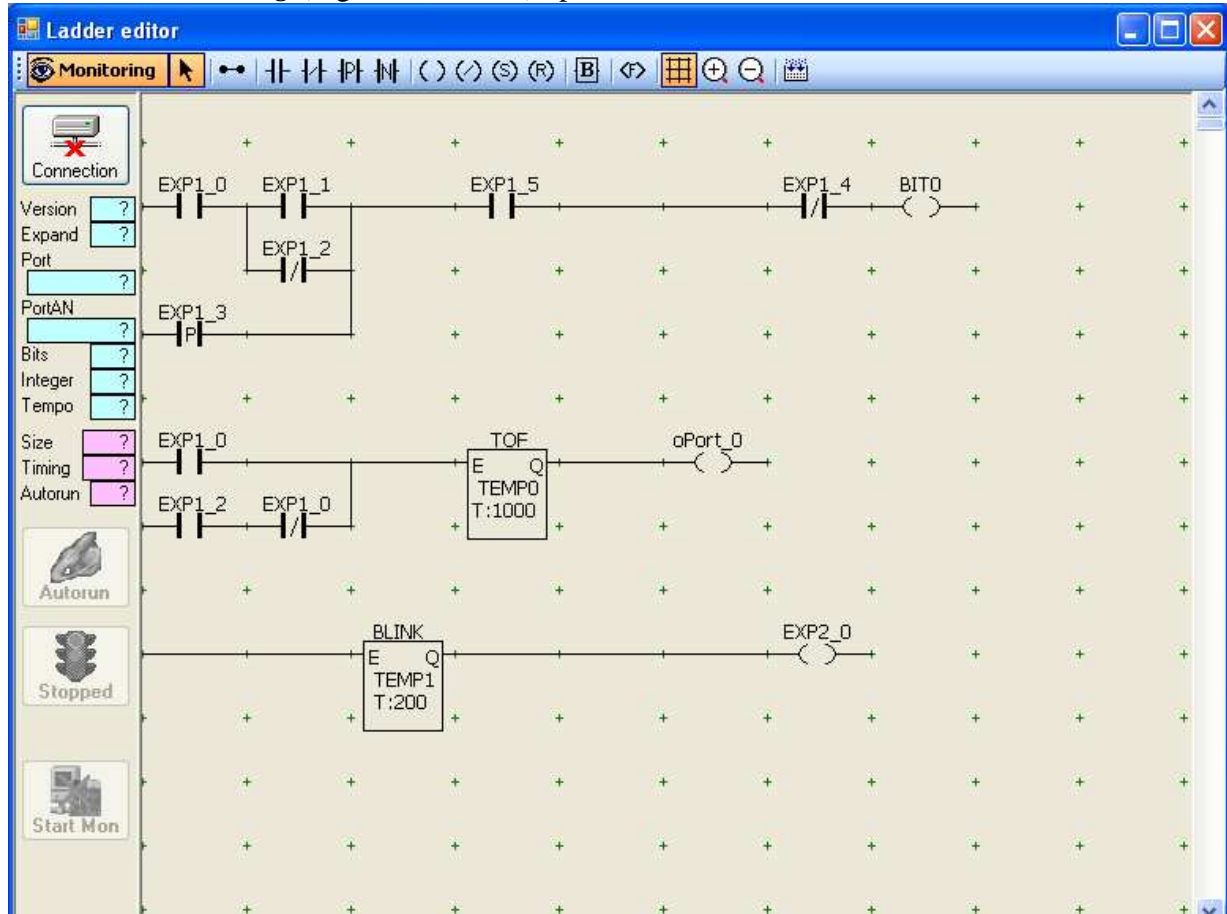


Figure 19

When connecting PC ⇌ DB, the target answers:

- 1/ Its version
  - 2/ The number of expanders which it possesses.
  - 3/ The mask bits of the main port (GPT0-3 on DB or port0-31 on ARM7)
  - 4 / The mask bits of PortAN used in IO TOR (0 on ARM7)
  - 5/ The number of bits of common usage.
  - 6/ The number of integers of common usage
  - 7/ The number of tempos
- \*\*\*\* if Fields 2 - 7 do not correspond to **Settings**, there is demand of update \*\*\*\*
- 7/ The size of PLC program loaded
  - 8/ The timing of execution of this one
  - 9/ If the program runs automatically after reset

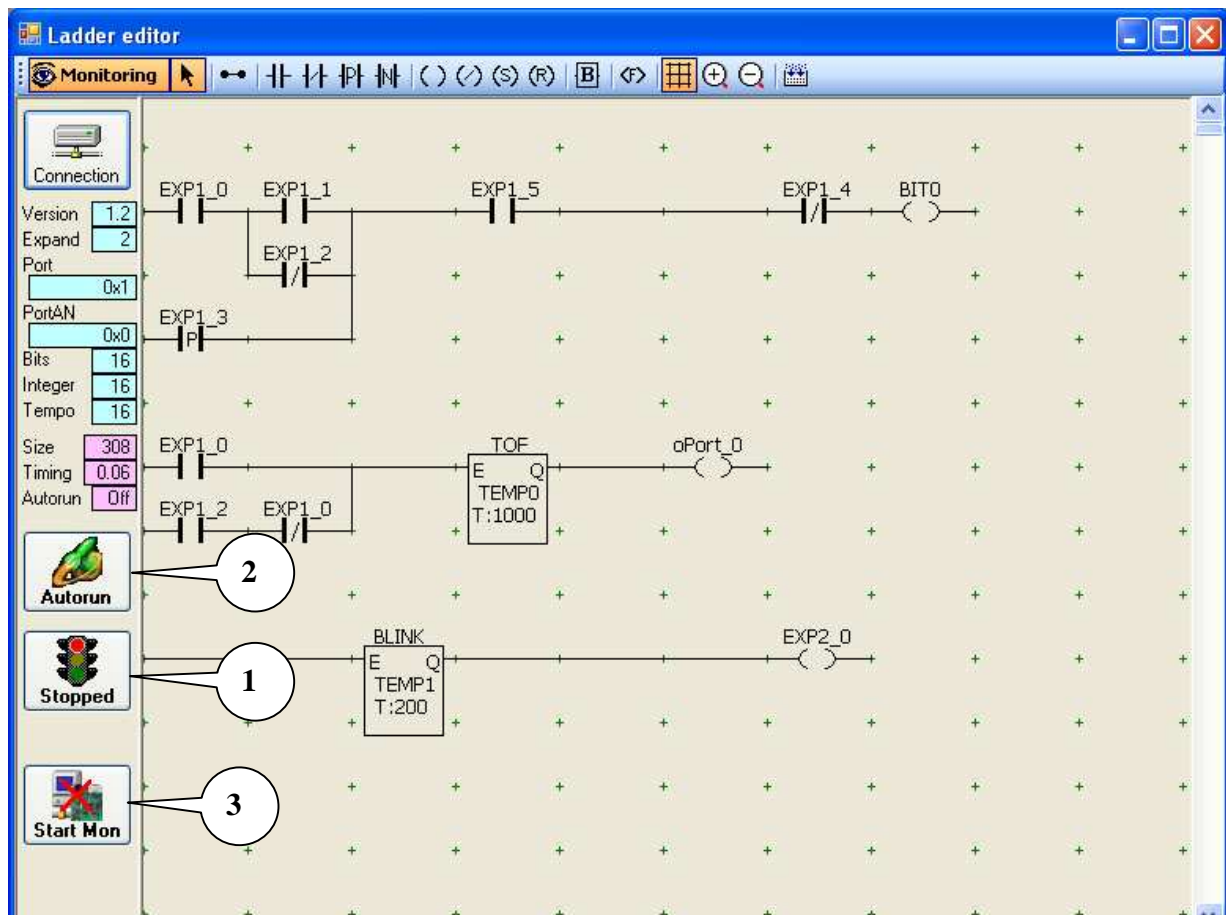


Figure 20

Button Figure 20 Field 1 activates or stops the PLC engine for debug.

After the program finalized, the Button **Aautorun** gives the property autorun to the PLC program. This autorun property is a line added to the file " PLC.TXT " by DB itself.

A click on Button **Aautorun** is necessary after a download (*Send to DB*) because the compilation does not set this property as active. In case there would be a blocking error (dead lock).

## Monitoring :

A click on **Start Mon** (Figure 20 Field 3) establishes the communication DB with the target (DB or ARM7).

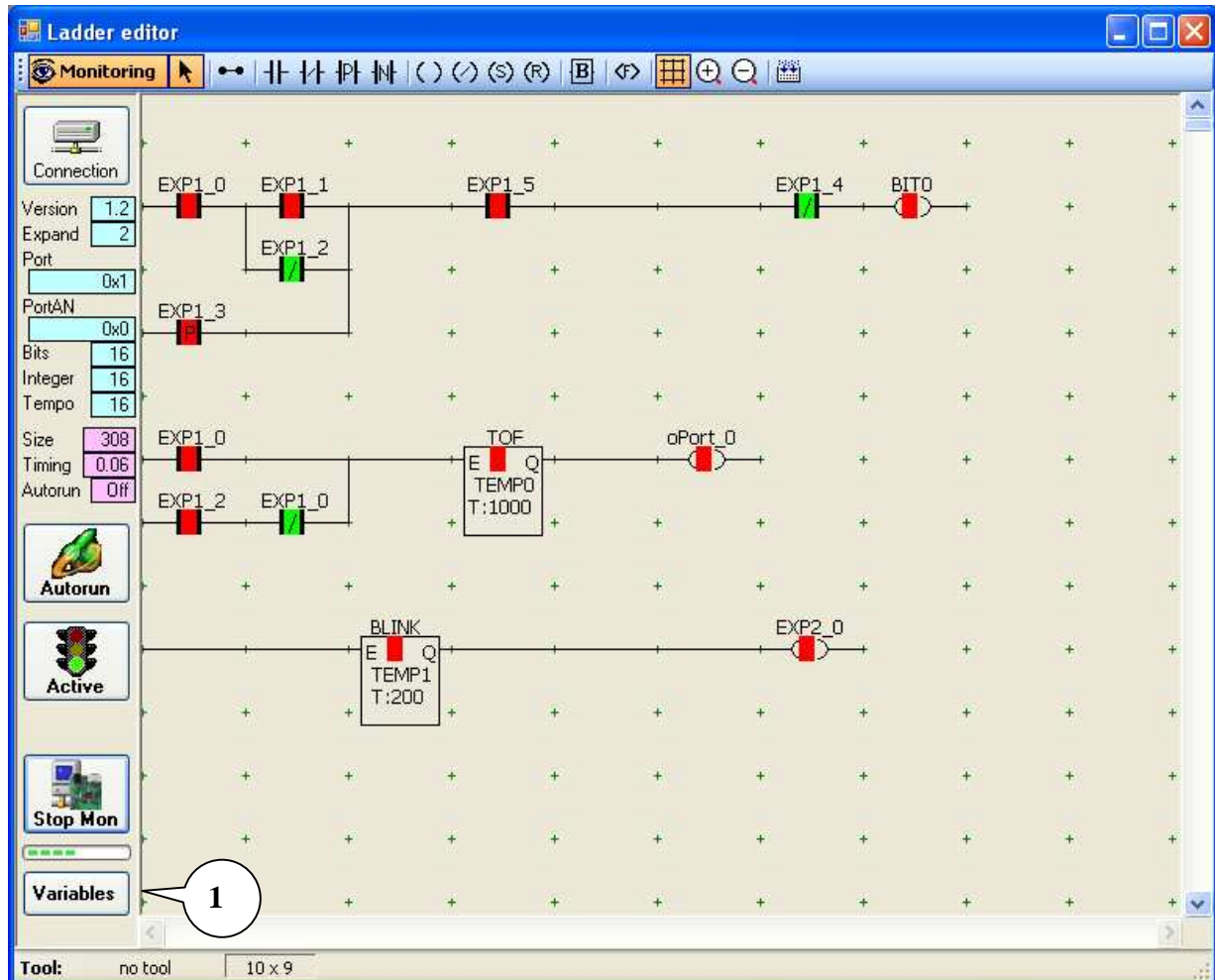


Figure 21

When we are in monitoring, we cannot modify any more the graph but only move.

The program questions DB about the state of objects on the sheet.

When the response is received, the graph is animated with green and red according to the state of the objects.

Button **Variables** appears (Figure 21 Field 1).

A click on **Variables** shows a zone to display variables on the right of the sheet.

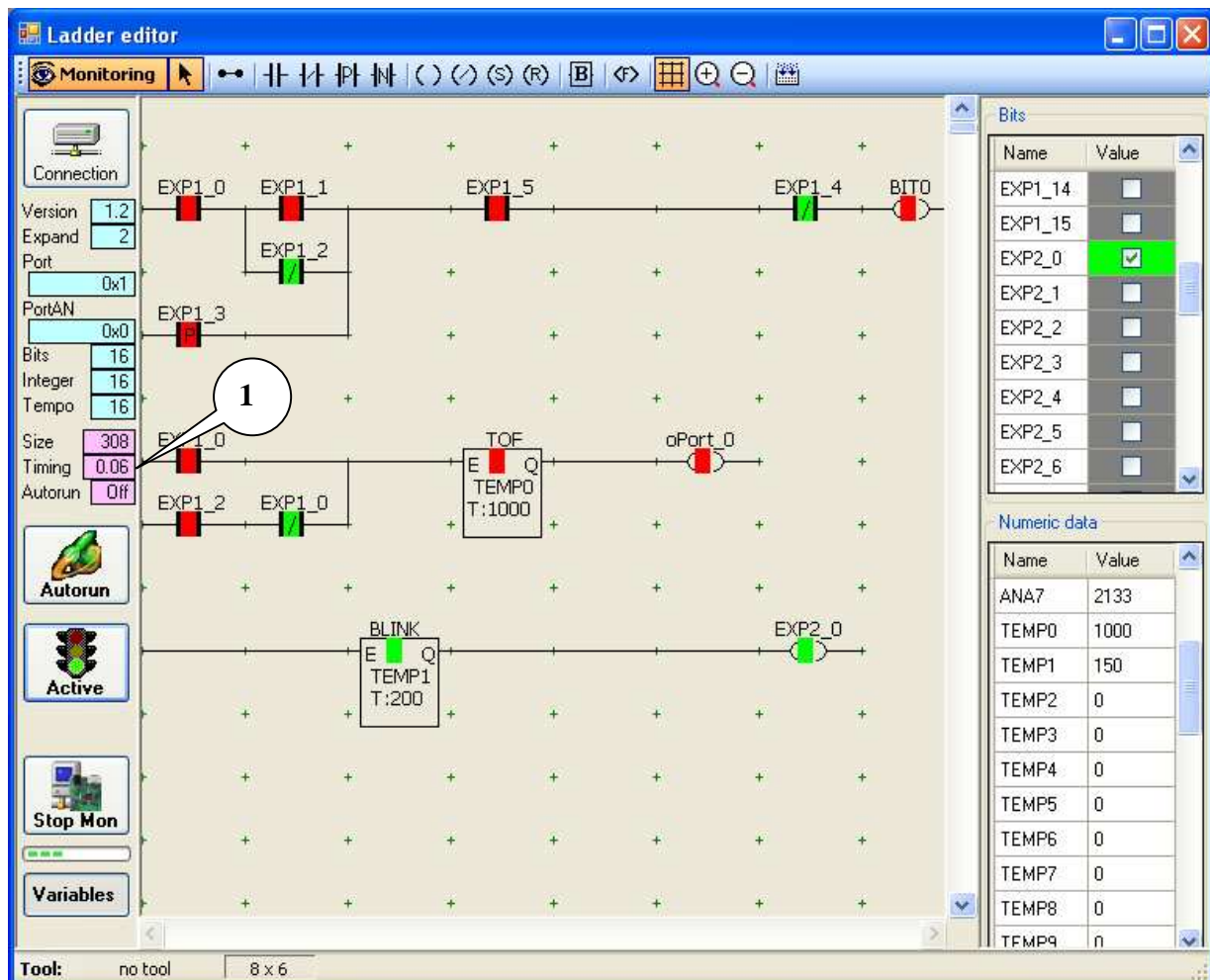


Figure 22

### Performances :

We can notice in the example Figure 22 field 1 that DB executes this program in 60 microseconds.



## Compatibilities Platforms Digibutler, ARMEE and Sceptre:

### **Differences**

	Digibutler (K3)	ARMEE	SCEPTRE
Microprocessor	ColdFire MCF52231 (59)	ARM7TDMI	ARM7TDMI
Quartz	25 MHz	14.745 MHz	12 MHz
Digital Ports	GPT0-3 et PortAN0-7	Port0-31	P0:0-31, P1:16-31
Analog Ports	PortAN0-7	-	AD0:0-7
Communication	Ethernet TCP/IP	Serial com 115 kbauds	Serial com Bluetooth
SD card	PortAN4-7	Port 18-21	SPI1
Expanders	optional GPT0-3	optional Port 3-7	optional I2C or SPI
RTC clock	Internal updated by SNTP	-	Internal
Compiler	CodeWarrior 7.1	WinARM-20060606	WinARM-20060606
Programmer $\mu$ P	BDM (TBLCF, P&E ...)	Serial (Flash utility)	Flash Magic

### **Communication PC $\Leftrightarrow$ Target**

Digibutler communicates in Ethernet TCP/IP while ARMY has only serial com ports.

Sceptre has only serial com ports on which Bluetooth (serial profil)

The communication PC  $\Leftrightarrow$  DB via several TCP ports was eliminated to offer to ARMEE the same features as DB.

### **Transfert files on SD card**

Tunnel FTP (client on PC and server on target) assures the interface.

The Tunnel FTP exists on 3 platforms.

DB possesses Tunnel FTP in a HTTP request.

ARMEE possesses a shell on the serial line for the monitor commands into which the Tunnel FTP is inserted ( Header+size+data+CRC16).

Sceptre possesses a shell on the serial com (USB) for the monitor commands.

Tunnel FTP is on serial com Bluetooth.

All the features of transfer PC  $\Leftrightarrow$  SD card are present on 3 platforms.

## ***Datalogger***

Not implemented on ARMEE (no RTC) or on Sceptre.

## ***PLC***

Thanks to the deletion of the multiple TCP connections, the majority of the functions PLC exist on 3 platforms.

ARMEE has no analog inputs but they appear in the list of choice ( future extension).

ARMEE does not measure the timing of the program and answers 0 ( future extension).

All other functions are operational with or without expander (extension board).

Sceptre has a 3-axis accelerometer (ana 0-2)

Sceptre has a thermometer (one wire). The temperature is stored in ana7 (not connected).